5Б

УДК 004.82:004.83

Є.В. Буров
Національний університет «Львівська політехніка» МОН України, м. Львів
Україна, 79013, м. Львів, вул. С. Бандери, 12

Менеджмент складних онтологій з використанням онтологічних моделей

Y.V. Burov
National university «Lvivska polytechnica» MES of Ukraine, c. Lviv
Ukraine,79013, c. Lviv, Bandery st., 12

Complex Ontology Management Using Ontological Models

Е.В. Буров
Национальный университет «Львовская политехника» МОН Украины
Украина, 79013, г. Львов, ул. С. Бандеры, 12

Менеджмент сложных онтологий с использованием онтологических моделей

У статті запропоновано використання онтологічних моделей задач для виконання завдань менеджменту складних онтологій. Зокрема, розглянуто завдання створення, модифікації, валідації та відслідковування походження елементів онтології. Визначено способи використання онтологічних моделей для вирішення цих завдань та переваги їх використання.
Ключові слова: онтологія, менеджмент онтології, онтологічна модель

In the article the usage of ontological models for facilitating the execution of various tasks of complex ontology management, such as ontology creation, modification, elements provenance tracking and validation is proposed. Methods for applying ontological task models to assist ontology management are developed and ensuing advantages are highlighted.
Key words: ontology, ontology management, ontological model.

В статье рассмотрено использование онтологических моделей для решения задач менеджмента сложных онтологий. Проанализирован способ использования этих моделей для создания и модификации, валидации, отслеживания истории изменений в онтологии предметной области. Определены преимущества использования онтологических моделей для решения задач менеджмента онтологий.
Ключевые слова: онтология, менеджмент онтологии, онтологическая модель.

Вступ

Центральним компонентом інтелектуальних, семантично-орієнтованих систем є онтологія – формальна, декларативна модель визначеної предметної області. Складність предметної області знаходить своє відображення і у складності відповідної онтології. Так, відомі онтології вищого рівня відображають значну кількість концептів, наприклад, CYC містить 2 мільйона, а Wordnet – близько 207 тисяч сутностей [1].

Складність створює суттєві проблеми у вирішенні задач менеджменту онтології. До цього класу задач відносять створення, оновлення, модифікацію, візуалізацію та валідацію онтології, документування походження її елементів. Відомо [2], що людина може тримати у фокусі уваги одночасно відносно невелику кількість об’єктів – 4 – 7.

Тому зі збільшенням розміру онтології, вирішувати задачі її менеджменту вручну стає усе важче. Одним з наслідків цього є протиріччя між кількістю сутностей (шириною онтології) та обсягом інформації, поданої для кожного елементу (глибиною онтології). Не дивно, що, наприклад, Wordnet на сьогодні використовують головним чином тільки як надзвичайно розвинений лінгвістичний ресурс – словник [1].

Складності у менеджменті онтологій у кінцевому результаті призводять до погіршення якості онтології. В [3] якість онтології визначається виконанням вимог щодо її повноти, коректності, стабільності. Помилки експерта підчас опрацювання складної онтології призводять до неврахування в онтології суттєвих концептів та зв'язків предметної області, що у свою чергу призводить до створення неповної та некоректної онтології.

Дослідження проблеми менеджменту складних онтологій проводиться в декількох напрямах. Зокрема, розробляють метрики та способи вимірювання складності онтології. В [4] по аналогії з визначеннями поняття складності програмного забезпечення, складність онтології визначена як труднощі у виконанні таких завдань, як розробка, повторне використання та модифікація онтології. В цій роботі розроблено багатовимірний набір метрик, які відображають як складність онтології загалом, так і її складність на рівні класів та відношень.

У роботі [5] запропоновано мета-онтологію O2, яка визначає онтологію як семіотичний об'єкт. Використовуючи цю онтологію розроблено три метрики складності онтології: структурні метрики, функціональні метрики та метрики профілювання придатності до використання (usability). Крім того, у цій роботі проаналізована значна кількість потенціальних метрик. Деякі з цих метрик є якісними і не можуть бути автоматично підраховані.

У роботі [6] запропоновані метрики для попередньо нормалізованої онтології. Нормалізація онтології включає такі кроки, як іменування класів, фактів, матеріалізація ієрархії наслідування, уніфікація імен, нормалізація атрибутів. Така нормалізація має на меті перетворення різних онтологій у семантично-еквівалентну форму з метою наступного створення семантичних метрик складності.

Функції менеджменту онтології інтегровані в інструментальні засоби розробки онтологій та концептуального моделювання, зокрема у такі, як Protégé [7] та TopBraid [8]. Існують системи менеджменту онтологій, орієнтовані на промислове використання [9].

Розробка засобів візуалізації онтології спрямована на підвищення ефективності роботи експерта з менеджменту онтологій. Робота багатьох цих засобів зводиться до відображення та уведення до фокусу уваги певної частини онтології, з якою працює експерт. Існуючі засоби візуалізації даних (Information Visualization) дозволяють застосувати візуальні метафори до визначеного набору даних, проаналізувати багатовимірні дані, часові зрізи та ін. Вони використовують комбінації текстового, табличного, діаграмного і графового відображення даних [10]. Інструментальні засоби для створення онтологій мають можливості візуалізації онтологій та фактів з інформаційної бази. Наприклад, Protégé має додаток OntoViz, який дозволяє відображати сутності та відношення онтології у вигляді графа, комерційний засіб для онтологічного моделювання TopBraid має можливості для відображення не тільки структури класів онтології, але й геоінформаційних даних.

Водночас, існуючі засоби візуалізації онтологій, наприклад, не дозволяють графічно відображати складні відношення, або ж відображати онтологію та факти у контексті задач, які вирішуються з використанням онтології.

Важливим завданням у менеджменті онтологій є відслідковування походження елементів та фактів онтології. Виконання цього завдання необхідне для валідації і забезпечення коректності онтології, адже предметна область змінюється і для підтрим-

ки коректності онтології необхідно відслідкувати залежності між елементами онтології, фактами інформаційної бази та відповідними об’єктами предметної області. На сьогодні визначено чотири рівня походження [11]: статичний (постійні дані), динамічний (змінні дані), нечіткий (походження цих даних по самій природі є нечітким, неясним), експертний (для отримання інформації про походження необхідна оцінка експертів). У роботі [12] пропонується відслідковувати походження фактів шляхом запису історії їх зміни, а також детальнішого опису подій, що призвели до змін. Водночас, задача знаходження елементів схеми онтології які залежать від певного факту предметної області, у значній мірі залишається ще не вирішеною.

Значно легші у менеджменті є простіші за структурою онтології задач [13], що розглядають сутності та відношення предметної області в контексті рішення конкретних задач. Невелика складність онтологій задач дозволяє експерту охопити всі, релевантні для задачі, риси предметної області та відстежувати їх залежності, якщо предметна область змінилася.

Історично, онтології задач були запропоновані в результаті розвитку наукового напрямку аналізу задач (task analysis). Методи аналізу задач використовуються для визначення та формалізації усіх факторів, що впливають та використовуються в процесі вирішення задачі експертом. Такі методи широко застосовуються для проектування інтерфейсів комп’ютерних програм, в експертних системах, системах підтримки прийняття рішень [14].

Головним завданням цього напряму є аналіз та специфікація складових частин задачі, визначення її структури та обмежень. Це дозволяє експерту краще зрозуміти задачу, виявити свої помилки та упущення, змоделювати процес вирішення задачі та оцінити результати, передати свої знання іншим експертам та спроектувати комп’ютерний інтерфейс для вирішення задачі.

Значних змін напрямок аналізу задач зазнав з появою онтологій. Було запропоновано використання онтології задач (task ontologies) для формалізації концептів та відношень задачі [13]. На відміну від інших типів онтологій, таких як загальні або онтології предметних областей (general, domain ontologies), онтології задач:

– будуються окремо для класів подібних задач;
– важливим є концепт мети, пов’язаний з задачею, та його формалізація;
– вводиться концепт дії [15] та забезпечується виконання (або його моделювання) дії;
– реалізовано виконання моделі, побудованої на основі онтології задачі.

Дослідження онтологій задач тісно пов’язані з концептуальним моделюванням, адже в процесі побудови онтології задачі фактично створюється її формалізована концептуальна модель [16]. Важливим аспектом як концептуального, так і онтологічного моделювання задачі, є взаємодія з експертом предметної галузі, який створює та валідує онтологію.

У процесі досліджень онтологій задач були реалізовані середовища моделювання, які дозволяють створювати та виконувати онтологічні моделі для окремих класів задач. Найбільш розвиненим з таких середовищ є CLEPE (Conceptual level programming environment) [17].

На сьогодні головні дослідження в галузі онтологічного моделювання зосередилися на декларативних онтологіях – онтологіях предметних областей, загальних онтологіях [17]. Напрямок онтологій задач є недостатньо розвиненим.

З іншого боку, існуючі дослідження в галузі онтологій задач у значній мірі розглядають побудову онтологій для окремих класів задач. При такому підході виникають обмеження на переносимість та повторне використання онтологій для вирішення задач

в інших предметних областях, так як трактування одних і тих самих сутностей різними онтологіями задач буде різним.

Водночас, загальні онтології та онтології предметних областей формалізують значну кількість різноманітних концептів та відношень, які можна використовувати для побудови різних класів онтологій задач. Такі онтології задач будуть переносимими, адже вони базуються на спільній формалізації концептуалізації в онтології вищого порядка.

Онтологія задачі, побудована на основі сутностей і відношень загальної онтології, залежить від неї. Для відображення цієї залежності та уникнення двозначностей будемо називати онтології задач, побудовані на основі певної загальної онтології, онтологічними моделями, на відміну від незалежних онтологій задач.

Використання онтологічних моделей дозволяє спростити вирішення задач менеджменту складних онтологій.

Метою цієї роботи є визначення способів використання онтологічних моделей для спрощення вирішення задач менеджменту складних онтологій та підвищення їх якості.

Використання моделей задач для створення онтологій

Однією з проблем, з якою стикається розробник онтології визначеної предметної області, є багатоваріантність побудови онтології. Через складність та нечіткість предметної області, як правило, неможливо відобразити в онтології усі аспекти цієї області. На вибір концептів, які включають в онтологію, впливають як досвід розробника, так і його суб'єктивні уявлення про важливість тих чи інших концептів у предметній області. В результаті в онтологію можуть бути включені концепти, які не будуть ніколи використовуватися, а деякі важливі концепти, навпаки – не включені. Більш того, помилки, допущені підчас концептуалізації предметної області, можуть суттєво ускладнити подальший розвиток онтології. У цих умовах важливо визначити критерії прийняття рішення щодо включення концептів предметної області в онтологію.

Методології, що використовуються сьогодні для побудови онтологій [18], базуються на аналізі набору текстів з обраної предметної області для визначення головних концептів та відношень онтології. Такий підхід забезпечує відповідність онтології змісту предметній області, але відкритим при цьому залишається питання відповідності онтології комплексу задач, які вирішуються з використанням цієї онтології [19].

Подібна проблема в галузі проектування програмного забезпечення вирішується шляхом побудови та аналізу варіантів використання (use-cases) програмного продукту. Доцільно використати аналогічний підхід і до створення онтології – будувати онтологію у процесі аналізу комплексу задач, які вирішують з використанням онтології.

Процес створення онтології проходить для кожної задачі комплексу і складається з таких етапів:

– кожну задачу спочатку аналізують з метою визначення усіх релевантних для її вирішення концептів предметної області, а також відношень, обмежень та операцій;

– з використанням визначених компонент будують модель задачі. Форма подання моделі задачі орієнтована на використання її елементів в онтології. В першу чергу в моделі намагаються використати існуючі компоненти онтології, а потім додають нові, відсутні в онтології;

– оновлюють онтологію, додаючи до неї відсутні компоненти, виявлені у процесі аналізу задачі;

– проводять валідацію побудованої таким чином онтології з метою пошуку протиріч.

Таким чином, побудова спільної онтології відбувається в результаті побудови комплексу онтологій задач.

Будемо вважати компонентом онтології довільну її складову частину. Компоненти онтології утворюють множину CmOn, що належить до визначених базових типів даних:

$$CmOn = \{x \mid Type(x) \in \{T_{CL}, T_{SL}, T_{RSL}, T_{RU}, T_{CS}\}\},$$

де – T_{CL} тип даних для сутностей, T_{SL} – атрибутів, T_{RSL} – відношень, T_{RU} – правил, T_{CS} – обмежень. Функція Type(x) повертає тип даних компонента x.

Таким чином, аналізуючи компоненти онтології, що входять у модель задачі, можна визначити ті компоненти, які потрібно додати до онтології. При цьому до онтології додають такі компоненти Cm', для яких:

$$\begin{matrix} Cm' \notin T_{ON} \\ Cm' \in T_{MD} \end{matrix}.$$

Процес розробки та життєвий цикл онтології, побудованої з використанням онтологічних моделей, суттєво відрізняється від традиційного процесу побудови онтології. Загальна онтологія збільшується поступово, по мірі додавання нових онтологічних моделей. Кількість елементів та складність онтології зростає поступово, по мірі додавання до компетенції онтології нових задач (рис. 1).

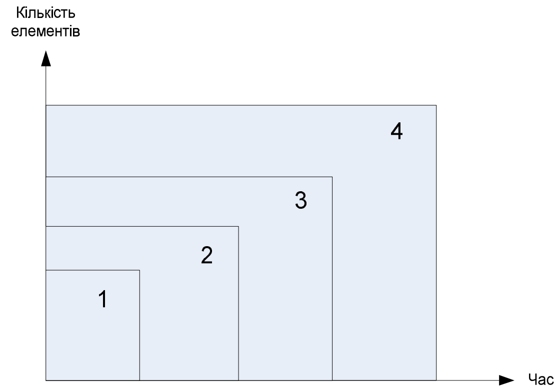


Рисунок 1 – Ітераційний процес розробки та розширення онтології

При цьому на кожному етапі створюється закінчена, валідована, придатна до використання та перевірена на вирішенні визначеного набору задач, онтологія. На кожному етапі виконується закінчений набір завдань по розробці онтології (специфікація, концептуалізація, формалізація, реалізація, інтеграція, валідація) [18].

Таким чином, суттєвими перевагами підходу з побудови онтологій на основі онтологічних моделей є спрощення процесу побудови за рахунок включення в онтологію мінімального набору елементів, необхідних для вирішення задач, концептуалізація на рівні задач, поступове зростання складності та проведення валідації онтології на кожному проміжному етапі, що в свою чергу приводить до підвищення якості онтології.

Модифікація онтологій

Предметна область, для якої побудовано онтологію, постійно змінюється. Важливим завданням менеджменту онтології є підтримка актуальності онтології, тобто відповідності актуальному стану предметної області. Це, у свою чергу, вимагає постійно аналізувати зміни у предметній області та модифікувати онтологію.

Іншою причиною модифікації онтології є зміна розуміння автором предметної області, вдосконалення або виправлення раніш зроблених в онтології помилок.

Для великих онтологій вирішувати постійно цю задачу доволі складно. Зокрема, не ясно, коли і як часто треба проводити аналіз предметної області та модифікацію онтології. Не просто відокремити від усіх змін предметної області зміни, релевантні для онтології, а також проаналізувати взаємний вплив цих змін та вплив на інші компоненти онтології. При цьому не до кінця зрозуміло, що є критерієм успішності впровадження змін.

Зазначені проблеми успішно вирішуються у системі, де онтологія доповнена множиною онтологічних моделей. Як правило, причиною запровадження змін в онтологію у такій системі є виявлення помилки або неточності, помічені при вирішенні задач з використанням моделей. У процесі аналізу помилкових моделей виявляють компоненти онтології, що потребують модифікації. Після впровадження змін в онтологію модифікують усі моделі, залежні від запропонованих змін. Критерієм успішності проведених змін є успішна валідація онтології. Модифікація онтології є ітераційним процесом. Знаходження помилок підчас валідації онтології вимагає перегляду запропонованих змін, їх модифікації. Таким чином, процес модифікації онтології буде складатися з наступних етапів:

– виявлення помилок у моделях, які вимагають змін в онтології;
– аналіз моделей та пропонування змін;
– запровадження змін в онтологію;
– валідація зміненої онтології;
– якщо у процесі валідації виявлені помилки, то повернутися на етап аналізу моделей, в яких виявлені помилки.

Нехай T^{i}_{MD} – онтологічна модель, в якій виявлені помилки, а On – онтологія. В результаті аналізу моделі T^{i}_{MD} визначають Cm^{i}_{MD} як множину елементів онтології, що змінюється. В результаті запровадження змін Cm^{i}_{MD} з'являється модифікована онтологія On'. Модифікована онтологія вважається прийнятною, якщо вона проходить валідацію, тобто

$$F_{val}(On') = true,$$

де $F_{val}(On)$ – функція валідації.

Відслідковування походження елементів онтології

Тісно пов'язаною з задачами модифікації та валідації онтології є задача відслідковування походження елементів онтології. У предметній області онтології часто існують факти, які визначають існування деяких елементів онтології (найчастіше – обмежень, або правил). Наприклад, такими фактами є законодавчі акти, що визначають вік повноліття, або пенсійний вік, межу бідності тощо. Втрата актуальності таких фактів вимагає проведення змін в онтології.

У ширшому сенсі задачу відслідковування походження розглядають як задачу збереження та аналізу історії створення та змін елементів онтології. Зокрема у [12] пропонується фреймворк, який дозволяє для кожної зміни занотувати сім її визначних аспектів (хто, що, де, коли, чому, як, який).

Включення у базу знань комплексу онтологічних моделей дозволяє додати нові можливості для вирішення задачі походження елементів онтології. Зокрема,

– в процесі створення моделі природньо виявити усі факти, що впливають на виконання, а потім доповнити онтологію посиланням на ці факти;

– у процесі валідації моделі відбувається і валідація посилань на факти предметної області, які вона використовує;

– використання моделей дозволяє проаналізувати ступінь впливу конкретного факту на комплекс моделей, і, таким чином визначити важливість цього факту;

– виникає можливість проведення аналізу історії змін онтології у розрізі онтологічних моделей. До семи аспектів кожної зміни в [12] доцільно додати ще назву моделі (задачі) модифікація якої послужила причиною зміни онтології;

– використання онтологічних моделей у базі знань дозволить вирішувати задачі прогнозування змін у результатах виконання моделей при зміні фактів предметної області, на які посилаються моделі.

Визначення якості та валідація онтологіїї

Якість онтології визначається через дотримання таких вимог, як вимоги до повноти, коректності та відсутності надлишковості [3]. Зокрема, вимога повноти формулюється так: «Усі релевантні для визначеного кода задач аспекти предметної області повинні бути відображені в онтології» [3]. Онтологія вважається коректною, якщо знання, визначені у ній, коректні для визначеної предметної області та релевантні для функцій, які виконує інтелектуальна система, що використовує онтологію.

Якість онтології має декілька аспектів. Зокрема, в [5] визначені структурні, функціональні та орієнтовані на користувача аспекти якості. Структурні аспекти якості відслідковують, наприклад, правильність побудови таксономії сутностей онтології. Функціональні відображають придатність онтології для вирішення комплексу задач (функцій) онтології, а користувацькі – зручність роботи кінцевого користувача. Перевірка деяких аспектів якості, наприклад, структурних, може бути проведена машинно. З іншого боку, перевірку функціональних аспектів найчастіше проводить експерт – автор онтології [5]. Оцінку якості онтології доцільно розділити на верифікацію (перевірку формальної, структурної цілісності онтології) та валідацію (перевірку функціональних аспектів онтології).

У випадку складної, загальної онтології з нечітко визначеним набором застосувань провести валідацію онтології та довести її повноту та коректність складно, адже об'єктивні критерії релевантності та коректності відсутні. На практиці створену онтологію валідує її автор, визначаючи набір релевантних на його думку концептів та відношень предметної області, які необхідно включити в онтологію.

Використання онтологічних моделей дозволяє довести коректність та повноту загальної онтології через доведення коректності та повноти комплексу онтологічних моделей, побудованих на основі цієї онтології. Онтологічна модель, порівняно з загальною онтологією, містить набагато менше елементів, вона проста у розумінні для автора-експерта. І тому процес її валідації простіший та ймовірність виникнення помилки менша, порівняно з аналогічним процесом для загальної онтології.

За аналогією з визначеннями для онтології будемо вважати **повною** модель, якщо вона містить усі необхідні для вирішення поставленої задачі сутності, відношення, обмеження та операції.

Будемо вважати, що модель **не містить надлишкових елементів**, якщо всі її елементи використовуються для вирішення задачі і вилучення хоч би одного з них унеможливить коректне вирішення задачі. Будемо вважати **коректною** модель, яка вирішує поставлену задачу згідно із заданими критеріями ефективності.

Нехай для заданої предметної області визначено множину повних та коректних моделей $M(T_{MD})$. Тоді онтологія, побудована на основі цієї множини моделей, є повною, якщо:

$$\forall Cm_i \in On : Cm_i \in T_{MD}^i \in M(T_{MD}).$$

Онтологія не є надлишковою, якщо усунення довільного її елементу унеможливить коректне вирішення задач у системі. Наявність онтологічних моделей полегшує пошук надлишкових елементів в онтології.

Так, довільний елемент CM_i, який використовується в моделі Md_j, не є надлишковим. Це дозволяє зменшити розмір множини потенційно надлишкових елементів онтології, які потрібно проаналізувати підчас аналізу онтології на надлишковість. Онтологія є **коректною,** якщо вона побудована на основі коректних моделей.

Додатковою і суттєвою перевагою використання онтологічних моделей задач для валідації онтології є можливість виявити всі недоліки у процесі практичної експлуатації онтології – вирішення комплексу задач, поданого онтологічними моделями.

Висновки

Отже, використання онтологічних моделей, побудованих на основі спільної загальної онтології в єдиній базі знань, дозволяє спростити вирішення задач менеджменту складної онтології, зокрема в аспектах її створення, модифікації та валідації. При цьому загалом підвищується якість онтології, зменшується кількість помилок концептуалізації, а сама онтологія валідується та удосконалюється у процесі її практичного використання.

Литература

1. Medelyan O. Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense / O. Medelyan, Legg // Proceedings of the WIKIAI Wikipedia and AI Workshop at the AAA. – 2008. – Vol. 8. – P. 13-18.
2. Гиппенрейтер Ю. Психология внимания / Ю. Гиппенрейтер, В. Романов. – М. : Че-Ро, 2001.
3. Gomez-Perez A. Evaluation of ontologies / A. Gomez-Perez // International Journal of intelligent systems. – 2001. – Vol. 16, issue 3. - P. 391-409.
4. Zhe Y. Evaluation Metrics for Ontology Complexity and Evolution Analysis / Y. Zhe, D. Zhang, Y.E. Chuan // 2006 IEEE International Conference on eBusiness Engineering ICEBE06. – 2006. – P.162-170.
5. Gangemi A. A theoretical framework for ontology evaluation and validation / A. Gangemi, C. Catenacci, M. Ciaramita, J. Lehmann // Semantic Web Applications and Perspectives SWAP2nd. – 2005.
6. Vrandecic D. How to Design Better Ontology Metrics / [D. Vrandecic, Y. Sure, E. Fraconi and eds.] // Lecture Notes in Computer Science The Semantic Web Research and Applications. – Springer, 2007. – P. 311-325.
7. The Protégé Ontology Editor and Knowledge Acquisition System [Electronic resource]. – Access mode : http://protege.stanford.edu.
8. Top Braid Composer [Electronic resource]. – Access mode : http://www.topquadrant.com/products/TB_Composer.html.
9. Das A. Industrial Strength Ontology Management / Das A., Wu W., McGuinness D.L. – Stanford University, 2001. – 101 p.
10. Lanzenberger M. Visualization in Ontology Tools / Lanzenberger M., Sampson J., Rester M. // 2009 International Conference on Complex Intelligent and Software Intensive Systems, 2009. – P. 705-711.
11. Huang J. Dynamic Knowledge Provenance / J. Huang, M.S. Fox // Proceedings of Business Agents and Semantic Web. – 2003. – P. 1-18.
12. Liu Jun. Understanding the semantics of data provenance to support active conceptual modeling / Liu Jun // Active conceptual modeling of learning. – 2007. – P. 1-29.
13. Van Welie Martin. An Ontology for Task World Models / [Van Welie, Martijn, Van Der Veer, Gerrit C Eliëns] // Methods. – 1998. – V. 98. - P. 57-70.
14. Johnson Peter. Task-Related Knowledge Structures : Analysis, Modelling and Application / [Johnson Peter, Johnson Hilary, Waddington Ray, Shouls Alan] // Knowledge Creation Diffusion Utilization. – 1988. – P. 35-62.

15. Raubal Martin. Ontology-Based Task Simulation/ M. Raubal, W. Kuhn // Spatial Cognition and Computation. – 2004. – V. 4. – P. 15-37.
16. Seta Kazuhisa. Building ontologies for conceptual model management / [Seta Kazuhisa, Koyama Kazuya, Hayashi Yusuke, Ikeda Mitsuru] // WSEAS Transactions on Information Science and Applications. – 2006. – V. 3. – P. 546-553.
17. Mitsuri Ikeda. Task ontology: Ontology for building conceptual problem solving models/ [Mitsuri Ikeda, Kazuhisa Seta, Osamu Kakusho, Riichiro Mizoguchi] // Workshop note of application of ontologies and problem-solving meyhods, ECAI98, 1998.
18. A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies / Fernandez-Lopez ed. // OntoWeb. – 2002. – P. 56.
19. Brewster C. Data-driven ontology evaluation / [Brewster C., Alani H., Dasmahapatra S., Wilks Y.] // Computer. – 2004. – P. 164-169.

Literatura

1. Medelyan O. Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense / O. Medelyan, Legg // Proceedings of the WIKIAI Wikipedia and AI Workshop at the AAA. – 2008. – Vol. 8. – P. 13-18.
2. Gippenrejter Uj. Psihologija Wnimanija / Gippenrejter Uj., Romanow W. – M. : Che-Ro, 2001.
3. Gomez-Perez A. Evaluation of ontologies / A. Gomez-Perez // International Journal of intelligent systems. – 2001. – Vol. 16, issue 3. - P. 391-409.
4. Zhe Y. Evaluation Metrics for Ontology Complexity and Evolution Analysis / Y. Zhe, D. Zhang, Y.E. Chuan // 2006 IEEE International Conference on eBusiness Engineering ICEBE06. – 2006. – P.162-170.
5. Gangemi A. A theoretical framework for ontology evaluation and validation / A. Gangemi, C. Catenacci, M. Ciaramita, J. Lehmann // Semantic Web Applications and Perspectives SWAP2nd. – 2005.
6. Vrandecic D. How to Design Better Ontology Metrics / [D. Vrandecic, Y. Sure, E. Fraconi and eds.] // Lecture Notes in Computer Science The Semantic Web Research and Applications. – Springer, 2007. – P. 311-325.
7. The Protégé Ontology Editor and Knowledge Acquisition System [Electronic resource]. – Access mode : http://protege.stanford.edu.
8. Top Braid Composer [Electronic resource]. – Access mode : http://www.topquadrant.com/products/TB_Composer.html.
9. Das A. Industrial Strength Ontology Management / Das A., Wu W., McGuinness D.L. – Stanford University, 2001. – 101 p.
10. Lanzenberger M. Visualization in Ontology Tools / Lanzenberger M., Sampson J., Rester M. // 2009 International Conference on Complex Intelligent and Software Intensive Systems, 2009. – P. 705-711.
11. Huang J. Dynamic Knowledge Provenance / J. Huang, M.S. Fox // Proceedings of Business Agents and Semantic Web. – 2003. – P. 1-18.
12. Liu Jun. Understanding the semantics of data provenance to support active conceptual modeling / Liu Jun // Active conceptual modeling of learning. – 2007. – P. 1-29.
13. Van Welie Martin. An Ontology for Task World Models / [Van Welie, Martijn, Van Der Veer, Gerrit C Eliëns] // Methods. – 1998. – V. 98. - P. 57-70.
14. Johnson Peter. Task-Related Knowledge Structures : Analysis, Modelling and Application / [Johnson Peter, Johnson Hilary, Waddington Ray, Shouls Alan] // Knowledge Creation Diffusion Utilization. – 1988. – P. 35-62.
15. Raubal Martin. Ontology-Based Task Simulation/ M. Raubal, W. Kuhn // Spatial Cognition and Computation. – 2004. – V. 4. – P. 15-37.
16. Seta Kazuhisa. Building ontologies for conceptual model management / [Seta Kazuhisa, Koyama Kazuya, Hayashi Yusuke, Ikeda Mitsuru] // WSEAS Transactions on Information Science and Applications. – 2006. – V. 3. – P. 546-553.
17. Mitsuri Ikeda. Task ontology: Ontology for building conceptual problem solving models/ [Mitsuri Ikeda, Kazuhisa Seta, Osamu Kakusho, Riichiro Mizoguchi] // Workshop note of application of ontologies and problem-solving meyhods, ECAI98, 1998.
18. A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies / Fernandez-Lopez ed. // OntoWeb. – 2002. – P. 56.
19. Brewster C. Data-driven ontology evaluation / [Brewster C., Alani H., Dasmahapatra S., Wilks Y.] // Computer. – 2004. – P. 164-169.

RESUME

E.V. Burov

Complex Ontology Management Using Ontological Models

Ontology, being the central component of knowledge-based solutions, reflects the elements of domain. The inherent complexity of domain and struggle to make ontology feature-rich, including large number of domain characteristics, leads to creation of a complex, hard to manage ontology. On the other hand, smaller task ontologies are simple, they are easily created and validated manually by domain expert. In case, if task ontology is built using the elements of some larger domain (or general) ontology, such ontological model would present considerable advantages to assist the ontology management.

Ontology could be created as a result of development of large number of ontological models, thus sharing and reusing common elements of models in ontology. In software domain is common to utilize use-cases as a basis for software system creation. Analogically, ontological task models are creating use-cases allowing and constraining the building of a common ontology in ontology engineering domain.

Ontology domain as a part of a real world is constantly changing. Some of those changes does not affect ontology, but some should be introduced in ontology as soon as possible. In practice it is hard to track and analyze domain changes in order to modify ontology. The usage of ontological task models allows to quickly discover when and which modifications should be introduced in ontology.

Ontology is valid if it includes all relevant aspects of a domain, and is correct. In case of a large general-purpose ontology its validation is done by its author. There are no formal and objective criteria for validation. The usage of ontological models allows validating ontology through underlying ontological models. If all models are correct and valid, then common ontology is also valid.

Стаття надійшла до редакції 07.11.2012.